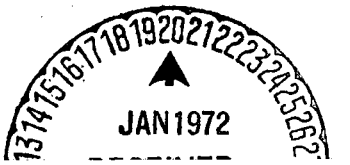
NASA TECHNICAL TRANSLATION NASA TT F-13,865

CODING A SEQUENCE OF DISCRETE READINGS OF MONOTONIC FUNCTIONS

A. B. Kryukov

Translation of "Kodirovaniye posledovatel'nosti diskretnykh otschetov monotonnoy funktsii". In Chetverty Simpozium po probleme izbytochnosti v informatsionnykh systemakh, Vol. II, Leningrad, 1970, pp. 388-395. (Fourth Symposium on the Problem of Redundancy in Information Systems)

13865



N72-14625 (NASA-TT-F-13865) CODING A SEQUENCE OF DISCRETE READINGS OF MONOTONIC FUNCTIONS A.B. Kryukov (Scientific Translation Service) Apr. 1971 10 p CSCL 12A
Unclas G3/19
(NA 12008
FACILITY FORM 602

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546 APRIL 1971

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U S Department of Commerce
Springfield VA 22151

CODING A SEQUENCE OF DISCRETE READINGS OF MONOTONIC FUNCTIONS

A. B. Kryukov

ABSTRACT. Data compression of a sequence of monotonic functions by a binary number sequence is treated. The contraction of the code increases with the number of data points.

In coding discrete functions, qualitative apriori data on their characteristic behavior are usually not taken into account. For example, the time sequence for readings to appear in an adaptive telemetric system is a discrete monotonic function. Bearing in mind such qualitative apriori data, we can code the functions more efficiently. In this paper we suggest two procedures for numbering all the monotonic functions to permit efficient coding of each with a uniform code. We analyze the effectiveness of this functional representation, and show particularly that with a fixed number of quantization levels the contraction coefficient grows with increase in the number of function readings. Characteristic of these procedures is the possibility of shaping a code word immediately upon arrival of the function reading to be coded. /388*

Let us look at a set of discrete functions, each of which is given by a sequence of readings $S = \{s_1, s_2, \ldots, s_i, \ldots, s_n\}$, where the readings s_i may assume any integral values from 1 to m. The number of possible functions is equal to m^n; therefore, to code any of them with an equal apriori probability that the functions will appear, it is necessary to expend $\langle \log_q m^n \rangle$ q digits. (Here and below, $\langle x \rangle$ denotes the least integer greater than or equal to x). In certain instances, we can superimpose qualitative limitations /389

*Numbers in the margin indicate pagination in the original foreign text.

on the behavioral characteristics of the functions. Then the number of possible functions is reduced and the coding procedure, for which $\langle \log_q m^n \rangle$ q digits are expended for each function, becomes redundant. If this same coding procedure is constructed by allowing for the limitations imposed on the functions to be coded, we can then reduce or totally eliminate the redundancy. In Reference [1], for example, a procedure is suggested for a redundancy-free code of strictly monotonic functions. In the present article, we suggest two procedures for redundancy-free coding of a wider class of functions, that are monotonic in the broad sense — that is, nondecreasing or nonincreasing. Below, for purposes of determinancy, we shall examine only the nondecreasing functions $s_1 \leq s_2 \leq \ldots \leq s_n$, although we can also code the nonincreasing functions, as well as both together, in an analogous manner. In the latter case, it is necessary to additionally expend one bit to indicate the monotonic character (nondecreasing or nonincreasing).

First Procedure. Coding is carried out in two stages. In the first stage the function is coded as a binary sequence of length m + n - 1; in the second stage, the sequence obtained is recoded into another, shorter binary sequence (on number). /390

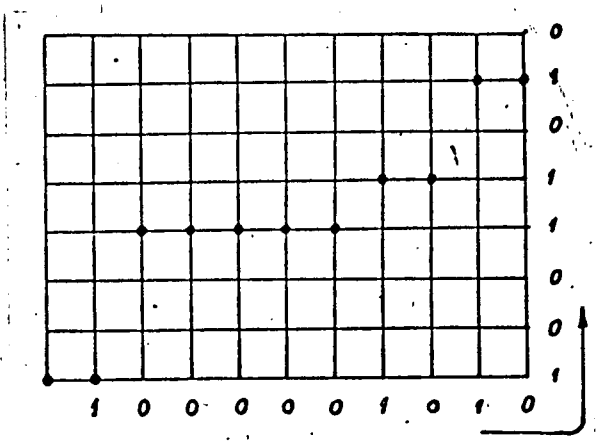
Let us make a sequence $a_1, a_2, \ldots, a_{m+n-1}$ of "zeros" and "ones" correspond to each nondecreasing function. It will consist of two parts. The first (n - 1) elements of the sequence are shaped by the rule:

$$a_i = \begin{cases} 1, & \text{if } s_{i+1} = s_i \\ 0, & \text{if } s_{i+1} > s_i \end{cases}, \quad i = 1,2,\ldots,n-1.$$

and the other m elements according to the rule:

$$a_{n-1+j} = \begin{cases} 1, & \text{if } j \in S \\ 0, & \text{if } j \notin S \end{cases}, \quad j = 1,2,\ldots,m.$$

Shaping of this sequence can be illustrated by the figure.



On the figure the first (n - 1) elements of the sequence are written in the form of a line, such that the i^{th} element of this part of the sequence corresponds to the $(i - 1)^{th}$ reading of the function. The other m elements of the sequence are written in the form of a column, such that the j^{th} element of this part of the sequence represents the j^{th} quantization level. The order /391 of numbering the elements in the sequence is shown by the arrow on the figure. In the line, the symbol "1" denotes those readings which are the same as the previous one; and in the column, the symbol "1" denotes those values which the function assumes.

Thus, each function is represented by a binary sequence of length m + n - 1, and the different functions are represented by different sequences which differ by the placement of their digits. The characteristic property of these sequences is that the number of "ones" in the sequence is always constant and equal to n. Correspondingly, the number of "zeros" is equal to m - 1. In fact, the first reading is always represented by a "one" in the column, and each subsequent reading is either the same as the previous one (represented in such case by a "one" in the line) or assumes a new value

(represented then by a "one" in the column). Proof of the one-to-one correspondence between the code functions and sequences may be based on the fact that the positions of the "zeros" in the line and the "ones" in the column are simply coordinates of the characteristic points (steps) of the function to be coded.

In the second stage, the sequence obtained is recoded into a shorter sequence (or number). The number of binary sequences of length $m + n - 1$, containing n "ones" is equal to C^n_{m+n-1}. Using the procedure suggested in [1], they can be recoded such that $\langle \log_q C^n_{m+n-1} \rangle$ q digits will be expended for each. For this, a certain number A corresponds to each sequence, wherein $0 \leq A \leq C^n_{m+n-1} - 1$. For formation of the numbers A, we use the number of those positions in the sequence where the "ones" (or "zeros") appear. Let us note that, generally, speaking, the numbers A may be formed both from the "zeros" and from the "ones" of the sequence since, $C^n_{m+n-1} = C^{m-1}_{m+n-1}$. The number of arithmetic operations in this case is proportional to the number of those symbols used. Therefore, the number of arithmetic operations can be made proportional to $\min(m - 1, n)$. /392

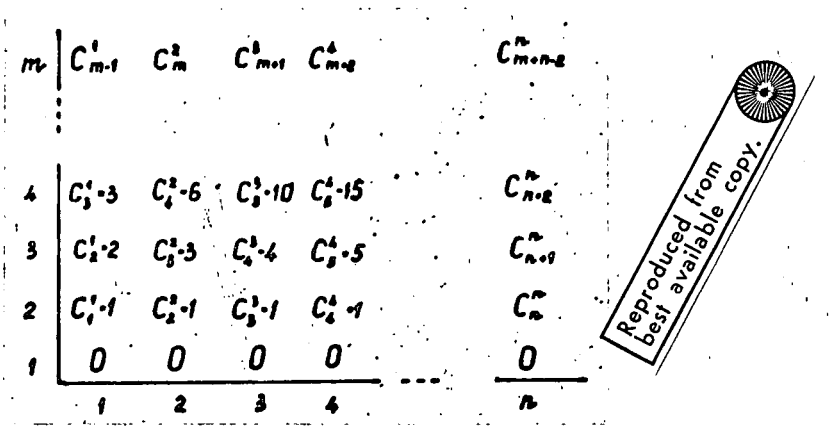
Second Procedure. Let us remember that in the first procedure a binary sequence of length $m + n - 1$ first corresponds to each function, and only then was this sequence coded. In the second procedure, the function is coded directly by elminating the formation stage of the sequence. For this, the functions are numbered, that is, a certain number B corresponds to each of them, wherein $0 \leq B \leq C^n_{m+n-1} - 1$. It is natural that the numbers B, representing different functions, are different. The rule for formation of these numbers consists of the following.

As before, let the function be given as a sequence of readings $S = \{s_1, s_2, \ldots, s_i, \ldots, s_n\}$. Then the number B for this function is defined as the sum

$$B = \sum_{i=1}^{n} C^{i}_{s_i + i - 2} \qquad (1)$$

where C^{β}_{α} is assumed equal to zero, if $\beta > \alpha$. Using the expression $C^{\beta}_{\alpha} = C^{\beta}_{\alpha-1} + C^{\beta-1}_{\alpha-2} + \ldots + C^{0}_{\alpha-\beta-1}$ it is easy to prove that, according to (1), different numbers of B represent different functions.

The procedure for forming the numbers B can be illustrated by using a special table consisting of the elements C^{i}_{k+i-2} ($i = 1, 2, \ldots, n$; $k = 1, 2, \ldots, m$):



With the exception of the lower line, consisting of individual zeros, this table is a Pascal triangle, in which the extreme left column of numbers C^{0}_{0}, C^{0}_{1}, ..., C^{0}_{m-1} is missing. Just like the Pascal triangle, this table may be constructed recurrently by using the expression $C^{\beta}_{\alpha} = C^{\beta}_{\alpha-1} + C^{\beta-1}_{\alpha-1}$. /393

The coding and decoding procedure using this table is analogous to the coding and decoding of the sequence of "zeros" and "ones" mentioned above. For example, let m = 4, n = 4 and the function be given by the readings {1, 3, 3, 4}. Then by "superimposing" the function, let us find on the table the number B as the sum of elements of the table, representing the readings of

the function, B = 0 + 3 + 4 + 15 = 22. This result naturally agrees with the value of B computed from (1). From the table it is clear that the minimal value of B is equal to zero when S = {1, 1, ..., 1}, and the maximal when S = {m, m, ..., m}, is equal to

$$B_{max} = C^{1}_{m-1} + C^{2}_{m} + \ldots + C^{n}_{(m+n-1)-1} = C^{n}_{m+n-1} - 1.$$

The rule for decoding consists of the following. In the first stage, we seek a value of k (k = 1, 2, ..., m), for which the inequality, /394

$$C^{n}_{k+n-2} \leq B < C^{n}_{k+n-1}. \qquad (2)$$

can be satisfied. Let (2) be satisfied when $k = s^{*}_{n}$. Then in the subsequent stage we seek a value of k (values of k exceeding s^{*}_{n} can be ignored), for which the inequality,

$$C^{n-1}_{k+(n-1)-2} \leq B - C^{n}_{s^{*}_{n}+n-2} < C^{n-1}_{k+(n-1)-1}. \qquad (3)$$

can be satisfied. Let (3) be valid for $k = s^{*}_{n-1}$. [In the Inequalities (2) and (3), just as in Equation (1), C^{β}_{α} is assumed equal to zero, if $\beta > \alpha$]. The procedure is continued until the sequence of values $S^{*} = \{s^{*}_{1}, s^{*}_{2} \ldots, s^{*}_{n}\}$ is reached. Let us note that, if at some stage of the decoding procedure — for example, when $k = s^{*}_{r}$ — the corresponding inequality becomes an equation, then all s^{*}_{i}, having $i < r$, can be immediately set equal to 1. The uniqueness of the decoding can be proven by using the known relationship $C^{\beta}_{\alpha} = C^{\beta}_{\alpha-1} + C^{\beta-1}_{\alpha-2} + \ldots + C^{0}_{\beta-\alpha-1}$. On the strength of the one-to-one agreement between the numbers B and the functions to be coded, S* = S.

Let us look at the decoding by using the table for this same example. In the first stage we seek in the fourth column the greatest number not exceeding B = 22. This number 15 is found in the fourth line. Let us compare the difference 22 - 15 = 7, and in the third column let us seek the greatest

number not exceeding 7. This number 4, found in the third line, means that $s_3 = 3$. Then let us compute the difference 7 - 4 = 3 and seek in the second column the greatest number not exceeding 3, and find the number 3 in the third line of this column. This means that $s_3 = 3$ and $s_1 = 1$, since 3 - 3 = 0. As a result of the decoding, we have obtained the original sequence of readings S = {1, 3, 3, 4}. /395

Estimate of the Effectiveness.

The procedures studied for coding nondecreasing functions permit us to expend $\langle \log_q C^n_{m+n-1} \rangle$ q digits for each of them. In the ordinary coding procedure, $\langle \log_q m^n \rangle$ q digits are expended. When m, n $\geq$ 2, $C^n_{m+n-1} < m^n$; therefore, the suggested coding procedures permit us to reduce the number of q digits to be expended. The advantage to be obtained — that is, the amount by which the number of q digits can be reduced — can be characterized by the contraction coefficient

$$K_{cont} = \frac{\langle \log_q m^n \rangle}{\langle \log_q C^n_{m+n-1} \rangle}. \qquad (4)$$

With a fixed m and $n \to \infty$, Expression (4) can be transformed into the form

$$K_{cont} \simeq \frac{n \log_q m}{(m-1) \log_q n} \cdot \left[1 + O\left(\frac{1}{\log_q n}\right)\right],$$

where

$$\lim_{\log_q n \to \infty} \left[\frac{O(1/\log_q n)}{1/\log_q n}\right] = \text{const.}$$

From (5) it is easy to see that the contraction coefficient grows infinitely with increase in n. Thus, the greater n, the greater is the advantage as a result of such coding of nondecreasing functions.

REFERENCES

1. Lynch, T. J. Sequence Time Coding for Data Compression. Proc. IEEE (letters), Vol. 54, October 1966.

Translated for National Aeronautics and Space Administration under Contract No. NASw-2035, by SCITRAN, P. O. Box 5456, Santa Barbara, California, 93103.